Design of Power System Simulation Software Based on Cloud Platform

Guanbiao Huang[1,*], Yinsheng Su[2,a], Chao Hong[3,b], Ligang Zhao[1,c], Xiaoshan Wu[1,d], Tinghui Zhou[1,e]

[1]Power System Research Department, CSG Electric Power Research Institute, Guangzhou, China
[2]Operation Mode Department, CSG Power Dispatching and Control Center, Guangzhou, China
[3]Leadership, CSG Power Dispatching and Control Center, Guangzhou, China

[*]260094093@qq.com, [a]suys@csg.cn, [b]hongchao@csg.cn, [c]zhaolg@csg.cn, [d]wuxs@csg.cn, [e]zhouth@csg.cn

Abstract. Aiming at the problem that DSP simulation software only supports C/S architecture simulation, three types of application service images are designed to deploy DSP simulation Software on the cloud platform. Based on Docker virtualization and containerization technology, cloud resources required for simulation are encapsulated into Docker containers. Users connect to the cloud using a B/S architecture, provide Docker containers, and directly use various resources allocated in the containers to complete modeling and simulation tasks in local browsers. The DSP simulation software based on cloud platform is a modular architecture that encapsulates simulation modules in functional packages. During simulation, specific functional packages are loaded using a hierarchical startup scheme based on task requirements.

Keywords: component; Cloud platform; Software Architecture; Docker; DSP

1 Introduction

DSP simulation software only supports the C/S architecture, which has higher security compared to the B/S architecture. Some special functions can only be implemented in the C/S architecture. However, the DSP software version varies in different operating systems, resulting in higher development and maintenance costs. It is necessary to install and configure the DSP simulation software operating environment on the client, resulting in poor distribution. During simulation, all built-in feature packages will be loaded at once, which has drawbacks such as high resource consumption and low simulation efficiency. DSP simulation software based on cloud platform [1] can support multi-user parallel simulation [2], with strong distribution and sharing, independent of operating system, and can achieve cross platform operation, with simple and convenient maintenance. DSP simulation

C. Chen et al. (eds.), *Proceedings of the 3rd International Conference on Digital Economy and Computer Application (DECA 2023)*, Atlantis Highlights in Computer Sciences 17,
https://doi.org/10.2991/978-94-6463-304-7_10

software utilizes Docker virtualization and containerization technology to virtual multiple Docker containers based on the number of users. Cloud resources can be allocated to Docker containers as needed. Users can connect to Docker containers through a B/S architecture to complete simulation tasks. During simulation, corresponding functional packages are launched according to simulation requirements, achieving on-demand allocation functions, solving problems such as high resource consumption and low simulation efficiency. DSP simulation software includes modules such as power flow calculation, electromechanical transient simulation, and short-circuit current calculation [3]. It is a modular architecture, with each type of function packaged into independent functional packages. The cohesion between functional packages is high, and the coupling is low. It can achieve service-oriented and object-oriented simulation functions.

To support users to use B/S architecture for DSP simulation, this paper designs three types of application service images to deploy DSP Software deployment on the cloud platform. Based on cloud platform hardware architecture and Software architecture, Docker virtualization and container technology are used to encapsulate simulation resources into containers to achieve Docker based B/S architecture simulation functions [4]. To address the issues of high resource consumption and low simulation efficiency under the C/S architecture, a hierarchical startup function package control system scheme is proposed to achieve simulation requirements with minimal resources.

2 Cloud Platform Architecture Design

2.1 Hardware Architecture and Network Environment

The cloud provider includes data clusters and computing clusters. Users can interact with the cloud platform through the B/S architecture without downloading DSP software [5]. By connecting to the Docker container through a local browser, they can access cloud platform resources and complete modeling and simulation tasks. The simulation results are graphical and output to the browser. During this process, the data cluster is responsible for storing calculation results and resource calls, implementing database management functions, while the computing cluster is responsible for parsing the model Compile source code, execute instructions, and complete scientific calculation.

Computing clusters provide services for user modeling, compilation, execution, and computing in the Platform as a service (PaaS) mode [6]. The data cluster utilizes the distributed system infrastructure Hadoop to configure on the cloud platform, providing data storage and management services for user simulation. Transaction data is managed by the relational database MySQL, program and model analysis results are stored and managed by HDFS during simulation, and calculation results and intermediate data are managed by the distributed storage system HBase.

2.2 Software architecture and Environment Configuration

The cloud platform software foundation is an efficient combination of various Open-source software such as "XenServer+Docker+Hadoop" [7][8]. The architecture has the characteristics of high cohesion and low coupling and can expand, and update hardware resources as needed. To reasonably allocate software and hardware resources and improve simulation speed, different levels of resource pooling and virtualization can be carried out according to different simulation links. The software system integrates IaaS (Infrastructure as a service), PaaS, and SaaS (Software as a service) architectures according to its own characteristics, and innovatively embed the DSaaS (Data Store as a Service) framework into the Software architecture [10].

Resource pools in cloud platforms are managed using XenServer virtualization and Docker container technology. XenServer is responsible for integrating computing resources, storage resources, and network resources into resource pools. The IaaS framework connects the hardware infrastructure of cloud platforms to various virtual containers or clusters through the network, and dynamically adjusts and manages the hardware infrastructure based on the hardware resources required for simulation tasks in the containers or clusters. The PaaS framework provides users with complete or partial software services, generates schedulers and computing cluster scheduling managers, and performs simulation function scheduling and calculations. The SaaS framework provides users with directly usable application software. DSP is a visualization software built by SaaS, and the SaaS framework can generate visualization demonstration schedulers, simulation inference schedulers, and simulation inference cluster scheduling managers. The DSaaS architecture ensures cloud security management from multiple dimensions, eliminates issues such as data leakage, abuse, and redundancy, and is the core framework of cloud platform simulation software. The functions of these frameworks are implemented based on Docker technology. Docker containerization technology integrates hardware facilities, and simulation instances built on cloud platforms can use hardware resources in parallel. The number of Docker virtual terminals, computing, and data resources can be dynamically adjusted according to the current server state and platform carrying capacity. Based on the priority of multiple users and tasks, various resources can be reasonably allocated to achieve the overall optimal solution. Real time, interactivity, parallelism, and visualization are external characteristics of cloud platform simulation software. SaaS utilizes Docker containers to manage multi-user parallel use of simulation software DSP for interactive visualization simulation. To ensure the compatibility of simulation software up or down, the data management cluster has designed many standardized interfaces to reduce module coupling, improve software portability, achieve plug, and play effects, and facilitate software updates and expansion.

3 Research on DSP Simulation Software

3.1 Overall architecture of DSP simulation software

DSP simulation software packages simulation modules in the form of functional packages into XCOS component disks, which have low coupling degree [9]. According to the role of functional packages in different stages of simulation and the scheduling depth of underlying core algorithms, functional packages can be divided into basic computing functional packages, core functional packages, and optional functional packages. These feature packages have a hierarchical startup control system solution. When new simulation requirements are submitted to the cloud platform, the DSP monitors the scheduling of software resources in real-time based on server feedback information, loads new feature packages for new tasks, satisfies simulation requirements with minimal resources, and does not affect simulation speed due to excessive activation of multiple feature packages.

The DSP core function pack is the foundation of various simulation functions, which is equivalent to the operating system during DSP simulation. It plays an important role in IO scheduling and coordination of hardware resources, handling interrupt events, process management, memory management, and file management. The basic computing function pack mainly implements simulation task calculation functions, and all computing clusters provide efficient computing services. The main computing capabilities of DSP are matrix operations and data compatibility operations. The optional feature pack is a user developed feature pack that, with the support of the first two feature packs, can be expanded as needed, applied reasonably, and completed complex task simulation functions.

3.2 Implementation of DSP Simulation Compilation and Running Functions

Compilation function implementation: DSP compilation is the process of integrating simulation models, checking compilation, and running environments, and initializing compilers.

The data generated by the simulation model integrated during DSP software compilation includes the number of module inputs and outputs, module size and appearance, module attribute data, interface data, port position data, block position data, etc. The model compilation function is implemented by writing functions in DSP language. The compilation function workflow involves initializing the compiler, compilation information, and required data for compilation, then obtaining global simulation settings data from the simulation framework, and finally generating simulation scheduling and status tables.

Implementation of operation function: During the DSP operation phase, simulation calculations are performed based on the compiled results, and the calculation results are presented to the user in a visual effect. In the DSP simulation run phase, simulation initialization, simulation Numerical integration, simulation status update, simulation numerical output calculation, simulation time output calculation, etc. are completed. The running process is to first receive the simulation compilation data transmitted by the DSP macro script, then call the simulation running function to execute the calculation function, select the simulation solver, set the parameters of the simulator, initialize the simulator, set the status of the calling module and the calculation function stage according to the current simulation time and the simulation state table information, and finally execute the corresponding calculation function of the module according to the simulation stage. The process flow of DSP software operation is shown in Figure 1.

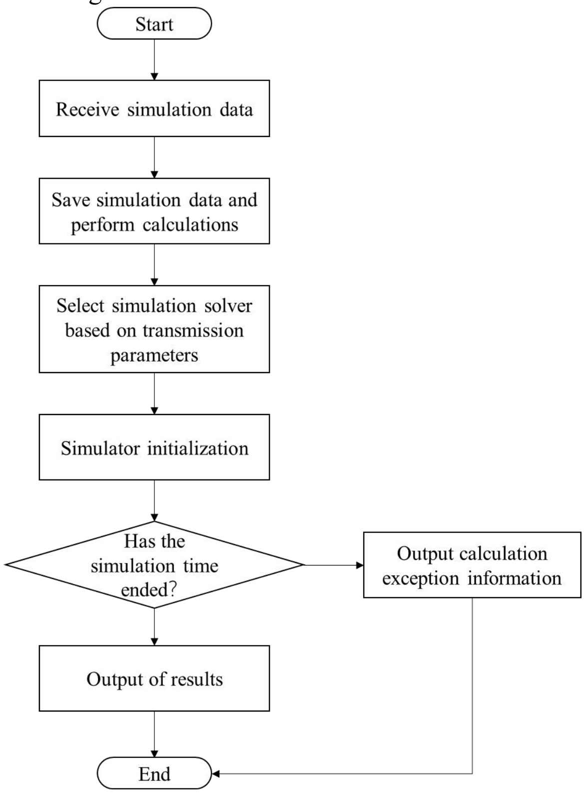

Fig. 1. DSP software operation flowchart

4 Implementation of underlying simulation software based on cloud platform

4.1 Implementation of Cloud Platform Software Architecture

The cloud simulation platform consists of eServer, Docker, Hadoop, and others. During the simulation process, the cloud platform creates a Docker container for each user, which is equivalent to a virtual computer. The container contains all the resources required for modeling and simulation. The data cluster and computing cluster adjust and manage various resources in real time. Users do not need to download DSP and can directly connect to the Docker container through a local browser to complete the simulation task, achieving a Docker based B/S architecture. Docker deploys simulation software functions and resource services, encapsulating the required resources and applications into Docker containers, and then porting the containers to computers that support Docker to achieve the function of "one encapsulation, run everywhere".

Resource scheduling and management in containers are achieved by Docker manipulating the cloud platform. Docker images provide static parameters such as underlying resources and libraries for Docker containers. There can be inheritance relationships between images, and base class images can derive sub images based on requirements. Generating a Docker container is the process of copying the Docker image and adding a writable layer. Subsequent operations are carried out in the container, and the resulting running results and status are recorded in the container through the writable layer.

The cloud platform provides three types of images to support the normal operation of DSP simulation, and web application service images that support video streaming facilitate user interaction with simulation software, presenting visual effects to users; The DSP basic computing application service image supports simulation calculation and deduction. After the simulation model is successfully built, submitting simulation commands can make the Docker container generated by this type of image work; DSP data supports application service image comprehensive management cloud platform, and the resources and data in the computing cluster and data cluster are allocated and managed by this type of image, which is equivalent to the operating system of the cloud platform.

4.2 Implementation of underlying functions of DSP simulation software

The underlying DSP uses interpretive language and relies on an interpreter, resulting in low efficiency but good cross platform performance. During the development of the function pack, DSP has developed a complete interface specification for the function pack. When loading and running the feature pack, DSP has designed two ways to implement it: one is to expose the Fortran interface to expand the new feature pack, and the other is to use DSP code to expand the new feature pack. When performing numerical operations such as matrix calculations, DSP code is used for implementation. When requiring fast execution speed or many loop and pointer operations, use an interface written in Fortran to complete.

The simulation software is internally equipped with a DSP function pool. After running the DSP function for the first time, information such as function name, function parameters, and return value will be saved in the DSP function pool, which accelerates the next call to the function and improves the simulation speed. When running Fortran functions, first compile them into the corresponding Dynamic-link library, and then load the dynamic library file into the DSP process. The DSP function pool will also save information about Fortran functions and Dynamic-link library files.

5 Conclusions

1) In the design of simulation software based on cloud platform, three types of application service images are provided to deploy DSP simulation Software deployment on the cloud platform, and the resources needed for simulation are encapsulated in the container by using Docker virtualization and container technology. Based on computing cluster and data cluster, simulation calculation and data management of simulation tasks are carried out through B/S architecture.

2) DSP simulation software is a modular core architecture, with different functional packages responsible for different functions at the bottom layer. They call and communicate with each other to complete simulation tasks together. During simulation, a hierarchical startup scheme is used to load corresponding functional packages and meet simulation requirements with the least resources, achieving the goal of saving resources and improving simulation efficiency.

3) The software structure of the cloud platform is based on the implementation of "XenServer+Docker+Hadoop". The three types of application service images support the normal operation and service of the cloud platform. DSP simulation based on the cloud platform is carried out on the B/S architecture of the three types of application service images and Docker implementation.

References

1. A. S D,Fernando F,Tetiana B, et al. ModelicaGridData: Massive power system simulation data generation and labeling tool using Modelica and Python[J]. SoftwareX,2023,21.
2. OGASAWARA Y,NISHI J,UDAGAWA Y, et al. Necessity and Targets of Future Electric Power System Simulation and Integration Studies for Realizing Higher Variable Renewable Energy Deployment and Lower CO2-Emission Society[J]. Grand Renewable Energy proceedings,2018,1(0).
3. Tang X W,Zheng Y W. Power System Simulation Data Description and Conversion[J]. Advanced Materials Research,2013,2584(765-767).
4. Zhao Y,Xie Y. Research on traffic scheduling based on private cloud platform[J]. Academic Journal of Computing & Information Science,2022,5(11).
5. Silva D E C,Diniz T,Cacho N, et al. Self-adaptive authorisation in OpenStack cloud platform[J]. Journal of Internet Services and Applications,2018,9(1).
6. Zhuo D,Yining G,Yi D, et al. A SCADA approach for Micro-grid based on Platform as a Service (PaaS) Delivery Model of Cloud Computing[J]. IOP Conference Series Earth and Environmental Science,2020,512(1).
7. Itaru K,Emi Y,Hitoshi O. Docker Vectorization, a Cloud-Native Privacy Agent—The Analysis of Demand and Feasibility for Era of Developing Complexity of Privacy Management[J]. Applied Sciences,2023,13(5).
8. Li Y,Hui S. Research and Design of Docker Technology Based Authority Management System[J]. Computational Intelligence and Neuroscience,2022,2022.

9. Purhonen A,Niemelä E,Matinlassi M. Viewpoints of DSP software and service architectures[J]. The Journal of Systems & Software,2004,69(1).
10. Diego J O,Marcos N,Paola C, et al. A real-time software framework for driver monitoring systems: software architecture and use cases[J]. REAL-TIME IMAGE PROCESSING AND DEEP LEARNING 2021,2021,11736.